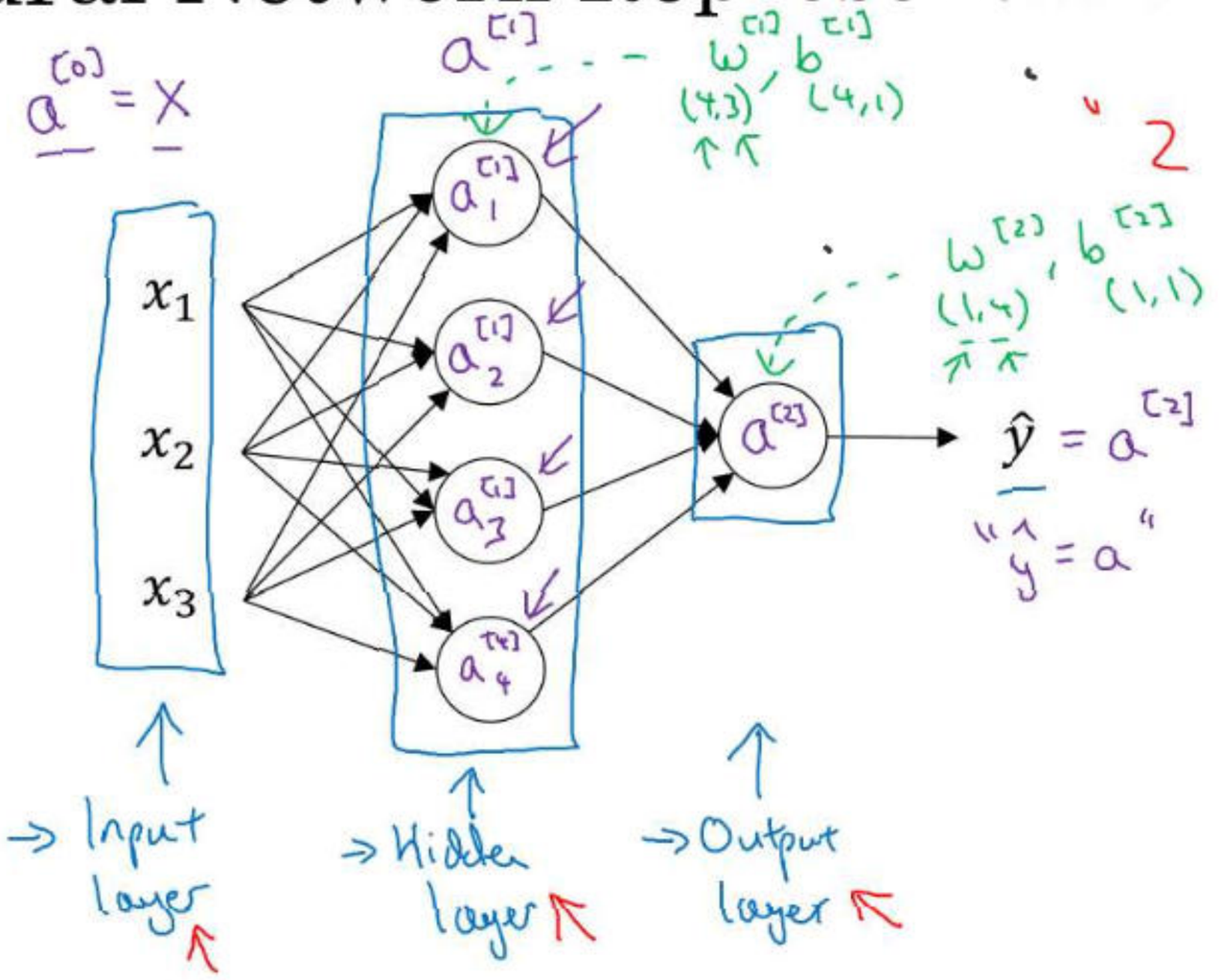
12 15

浅层神经网络推导.

Neural Network Representation



Andrew Ng

$$z_1^{[1]} = w_1^{[1]T} x + b_1^{[1]}$$

$$z_2^{[1]} = w_2^{[1]T} x + b_2^{[1]}$$

$$\vdots$$

$$\begin{bmatrix} - w_1^{[1]T} - \\ - w_2^{[1]T} - \\ \vdots \end{bmatrix} x + \begin{bmatrix} b_1^{[1]} \\ \vdots \\ b_n^{[1]} \end{bmatrix}$$

$$z^{[1]} = \begin{bmatrix} z_1^{[1]} \\ \vdots \\ z_n^{[1]} \end{bmatrix} \qquad z^{[1]} = w^T x + b.$$

$$a^{[1]} = \sigma(z^{[1]}).$$

$$z^{[2]} = w^{[2]T} \cdot a^{[1]} + b^{[2]}.$$

$$a^{[2]} = \sigma(z^{[2]}).$$

12 15. 多个实例：

每个实例 $x^{(i)}$，都会获得 $\Rightarrow a^{[2](i)} \longrightarrow \hat{y}$.

for i=1 to m:

$$z^{[1](i)} = W^{[1]} x^{(i)} + b^{[1]}$$

$$a^{[1](i)} = \sigma(z^{[1](i)})$$

$$z^{[2](i)} = W^{[2]} a^{[1](i)} + b^{[2]}$$

$$a^{[2](i)} = \sigma(z^{[2](i)})$$

令 $X = \begin{bmatrix} | & & | \\ x^{(1)} & \cdots & x^{(m)} \\ | & & | \end{bmatrix}$

$$A^{[1]} = \begin{bmatrix} | & & | \\ a^{[1](1)} & \cdots & a^{[1](m)} \\ | & & | \end{bmatrix}$$

↓ node.

$$Z^{[1]} = W^{[1]} X + b^{[1]}$$

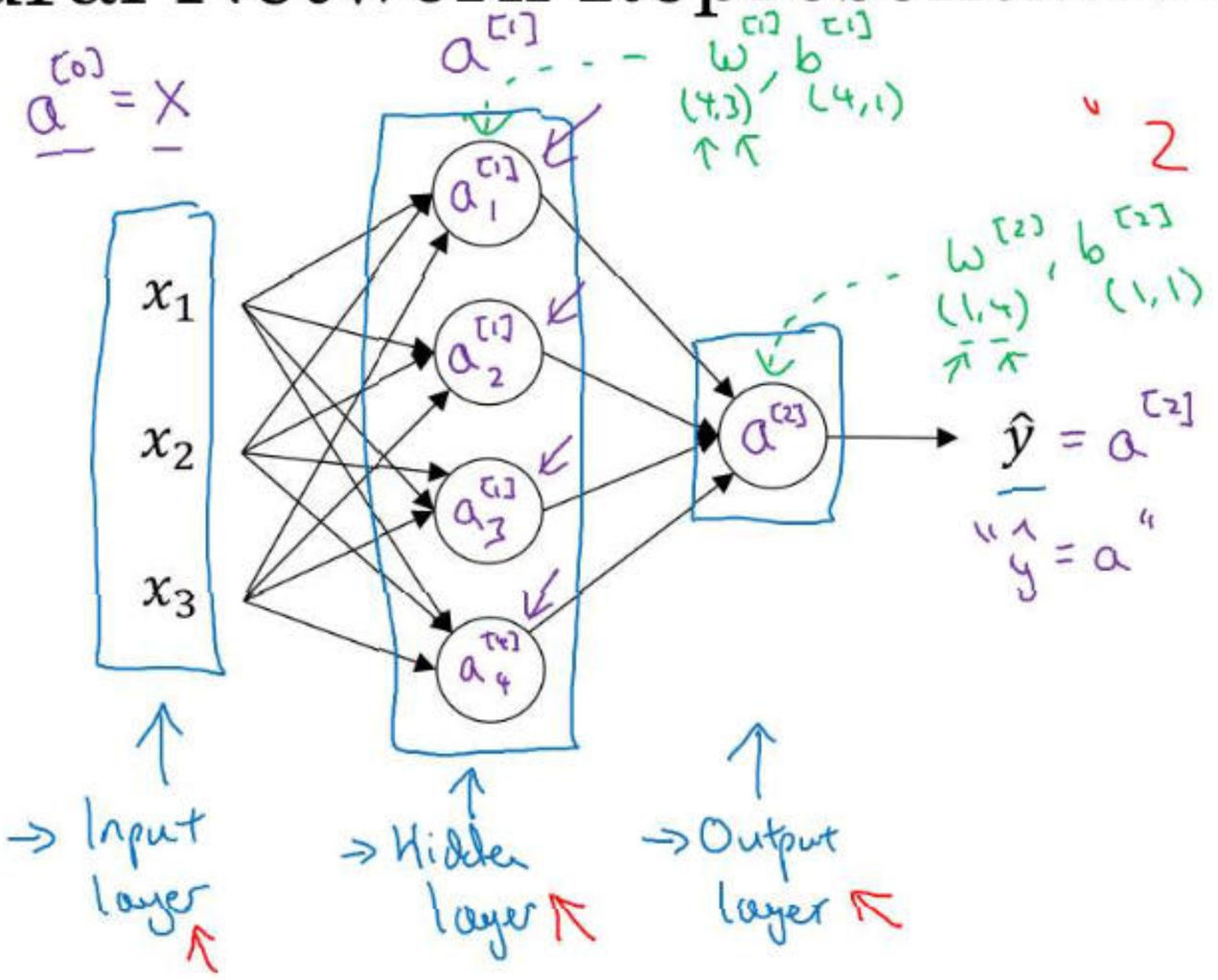
$$A^{[1]} = \sigma(Z^{[1]})$$

$$Z^{[2]} = W^{[2]} A^{[1]} + b^{[2]}$$

$$A^{[2]} = \sigma(Z^{[2]})$$

12 15. 梯度下降:

Neural Network Representation



Andrew Ng

网络参数: $w^{[1]}$, $b^{[1]}$, $w^{[2]}$, $b^{[2]}$.

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} L(a^{[2]}, y)$$

$\hat{y}^{(i)} = ?$

$$w^{[1]} = w^{[1]} - \alpha \cdot dw^{[1]}$$

$$\vdots$$

$$b^{[2]} = b^{[2]} - \alpha \cdot db^{[2]}.$$

$$dw^{[i]} = \frac{\partial J(\theta)}{\partial w^{[i]}}.$$

$$\left(\frac{1}{1+e^{-z}}\right)' = a(1-a)$$

$$dz^{[2]} = \frac{\partial L}{\partial a^{[2]}} \cdot \frac{\partial a^{[2]}}{\partial z^{[2]}} = -y(1-a^{[2]}) + (1-y)a^{[2]}$$

$$= a^{[2]} - y.$$

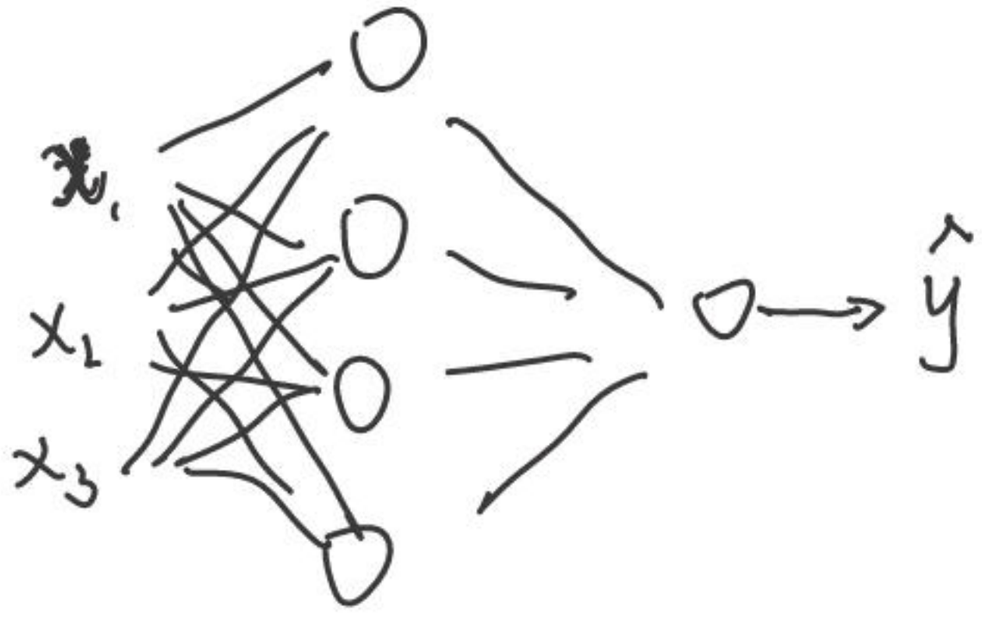
$$dw^{[2]} = dz^{[2]} \cdot \frac{\partial z^{[2]}}{\partial w^{[2]}} = dz^{[2]} \cdot a^{[1]T}.$$

$$db^{[2]} = dz^{[2]}.$$

12 15

$$da^{[1]} = \frac{\partial \mathcal{L}}{\partial z^{[2]}} \cdot \frac{\partial z^{[2]}}{\partial a^{[1]}} = dz^{[2]} \cdot w^{[2]}.$$

$$dz^{[1]} = da^{[1]} \cdot a^{[1]}(1-a^{[1]}).$$



两层有输出

一层 Hidden Layer.

12 16.

数据划分.

Train Dev / Valid Test.

数据量: 6:2:2. 大量 Dev + Test < 10%.

Dev, Test 分别有一定数量即可. Test 较少.

Some strategies.

a) Train 拟合不好:

调整模型, 构建更大 NN, 未收敛? 增加 Iter.

b): Dev 拟合不好:

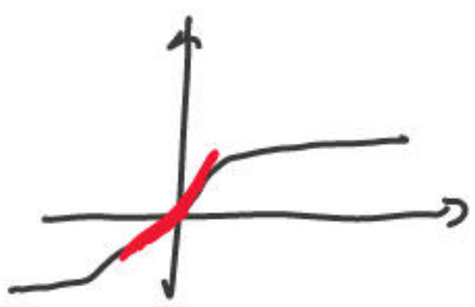
加 Reg; More Data.

Regularization:

减少过拟合的两种观点:

a): $\|W\|_F^2 \approx 0$, $\Rightarrow$ $z^{[l]} = W^{[l]} \cdot a^{[l-1]} + b^{[l]}$.

$z^{[l]} \approx 0$,



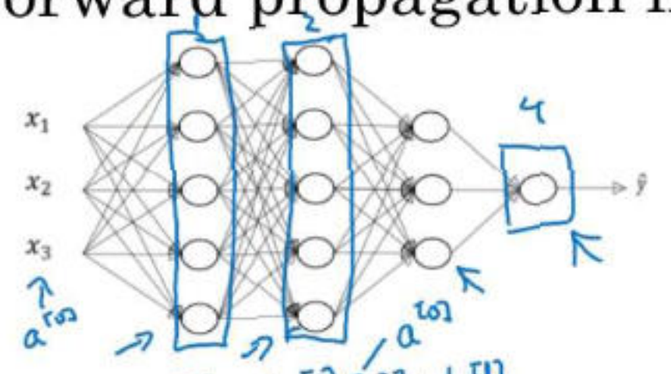
activate: $g(z)$ 近似线性, 线性 NN.

b): 神经元联结近似减少, 简单的 NN.

12 16

深层神经网络推导：

Forward propagation in a deep network



Andrew

前向传播：

$$z^{[1]} = \begin{bmatrix} z_1^{[1]} \\ \vdots \\ z_5^{[1]} \end{bmatrix} = \begin{matrix} w_1^{[1]} x + b_1^{[1]} \\ \vdots \\ w_5^{[1]} x + b_5^{[1]} \end{matrix} = W^{[1]} x + b^{[1]}$$

$a^{[2]} = g(z^{[2]})$.

向量化后：

$$z^{[2]} = \begin{bmatrix} | & & | \\ z^{[1][1]} & \cdots & z^{[1][m]} \\ | & & | \end{bmatrix} = \begin{bmatrix} | & & | \\ w^{[2]} x^{[1]} + b^{[2]} & \cdots & w^{[2]} x^{[m]} + b^{[2]} \\ | & & | \end{bmatrix}$$

$$= w^{[2]} \begin{bmatrix} | & & | \\ x^{[1]} & \cdots & x^{[m]} \\ | & & | \end{bmatrix} + \begin{bmatrix} b^{[2]} & \cdots & b^{[2]} \end{bmatrix} = w^{[2]} x + b^{[2]}$$

$$z^{[l]} = w^{[l]} A^{[l-1]} + b^{[l]}$$

$A^{[0]} = g(z^{[l]})$.

12 17

Backward propagation for layer l

→ Input $da^{[l]}$

→ Output $da^{[l-1]}, dW^{[l]}, db^{[l]}$

$dz^{[l]} = da^{[l]} * g^{[l]'}(z^{[l]})$

$dW^{[l]} = dz^{[l]} \cdot a^{[l-1]}$

$db^{[l]} = dz^{[l]}$

$da^{[l-1]} = W^{[l]T} \cdot dz^{[l]}$

$dz^{[l]} = W^{[l+1]T} dz^{[l+1]} * g^{[l]'}(z^{[l]})$

$dz^{[l]} = dA^{[l]} * g^{[l]'}(Z^{[l]})$

$dW^{[l]} = \frac{1}{m} dz^{[l]} \cdot A^{[l-1]T}$

$db^{[l]} = \frac{1}{m} np.sum(dz^{[l]}, axis=1, keepdims=True)$

$dA^{[l-1]} = W^{[l]T} \cdot dz^{[l]}$

Andrew Ng

反向传播：

$W^{[2]} \cdot a^{[1]} + b^{[2]} = z^{[2]}$ $\quad a^{[2]} = g^{[2]}(z^{[2]})$.

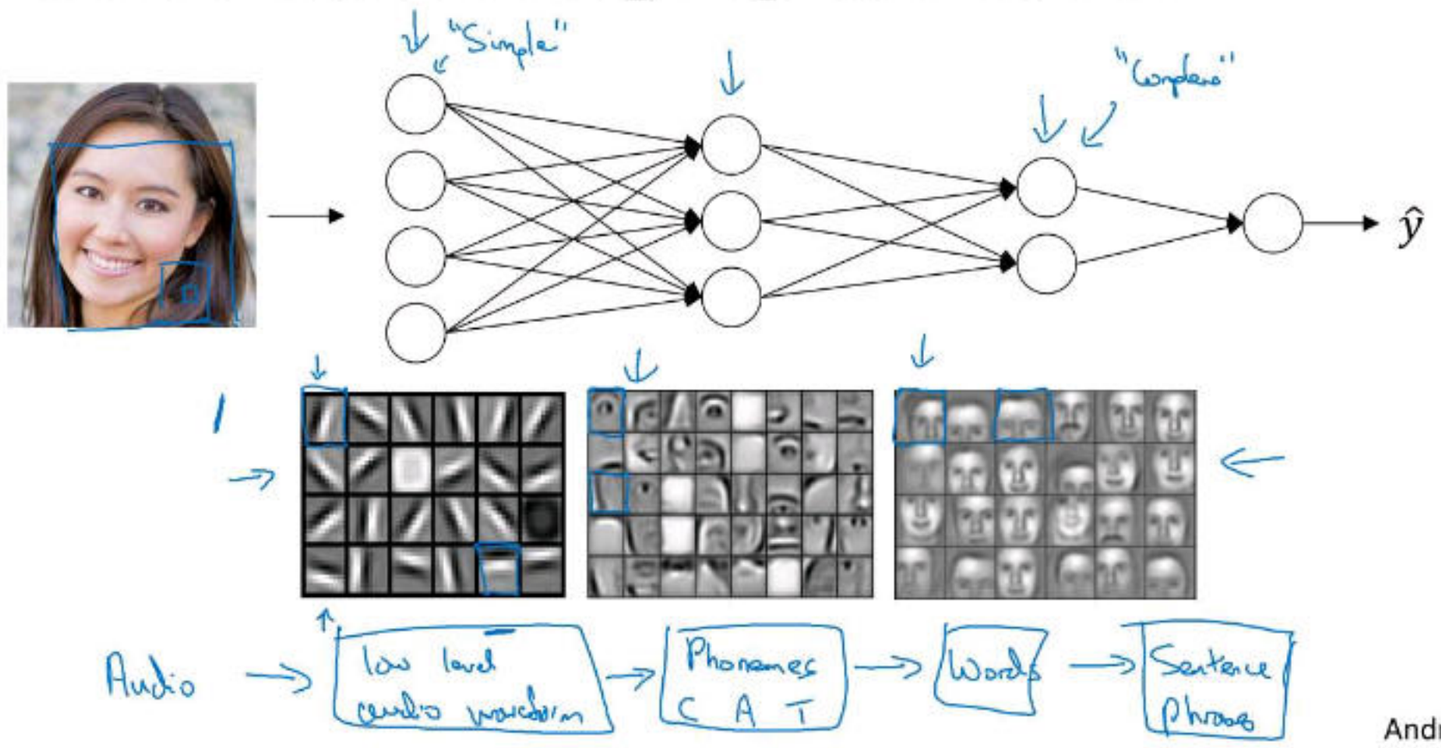
求导：$da^{[2]} = \frac{\partial L}{\partial a^{[2]}}$. $\quad da^{[1]} = \frac{\partial L}{\partial z^{[2]}} \cdot \frac{\partial z^{[2]}}{\partial a^{[1]}}$.

$\frac{\partial L}{\partial z^{[2]}} = \frac{\partial L}{\partial a^{[2]}} \cdot g^{[2]'}(z^{[2]}) = da^{[2]} * g^{[2]'}(z^{[2]})$.

$da^{[1-1]} = dz^{[2]} \cdot w^{[2]T}$.

12 17

Intuition about deep representation

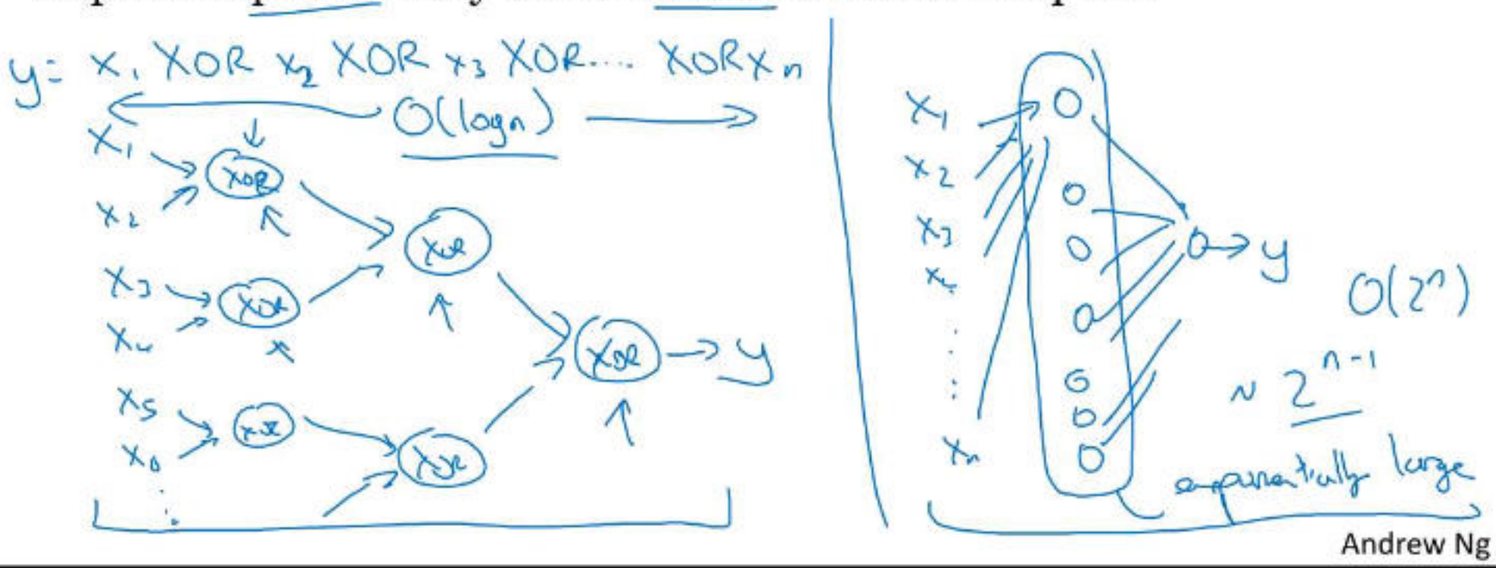


Andrew Ng

Some intuition: 隐藏层提取简单特征，组合成复杂特征，最后模式识别.

Circuit theory and deep learning

Informally: There are functions you can compute with a "small" L-layer deep neural network that shallower networks require exponentially more hidden units to compute.



Andrew Ng

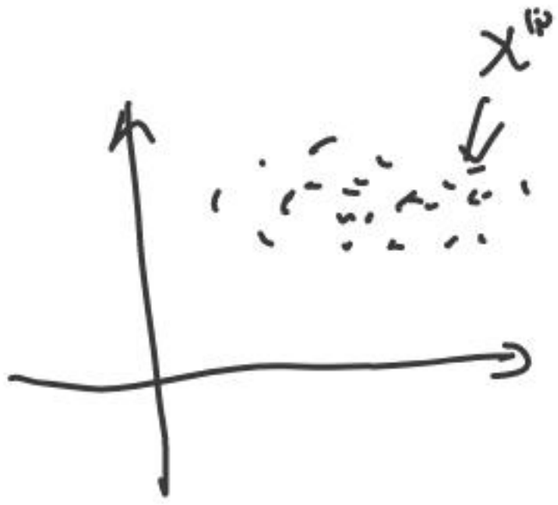
神经元越少（一个 hidden layer），需要网络越深.
若只有一层 hidden layer，神经元的数量会指数级增长.

12 17

避免过拟合的其它方法：

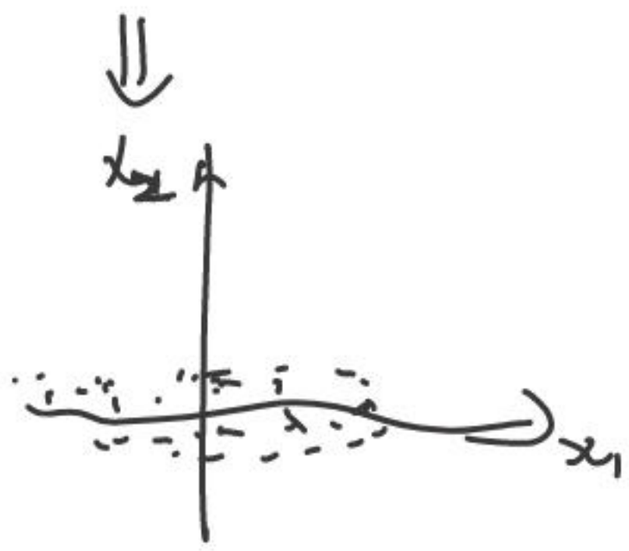
a): Data Augmentation: flipped, rotate ...
扩大了训练集.

b): Early stopping: 降低 dev Set Loss.



归一化特征.

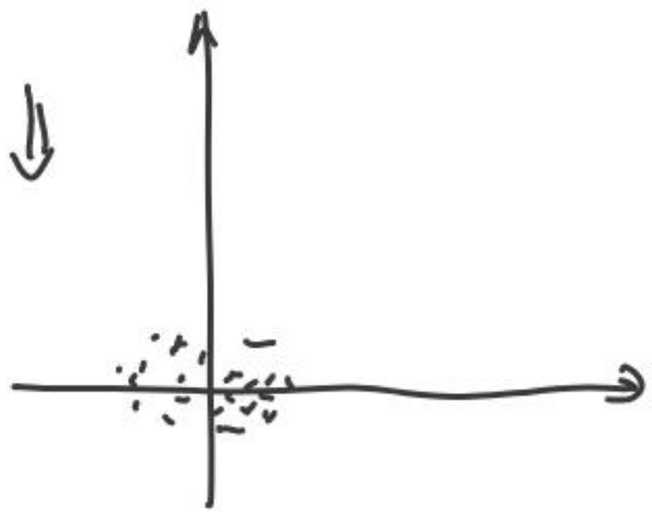
$$\mu = \frac{1}{m}\sum_{i}^{m} x^{(i)} \qquad x = x - \mu.$$



$$\sigma^2 x_1 > \sigma^2 x_2.$$

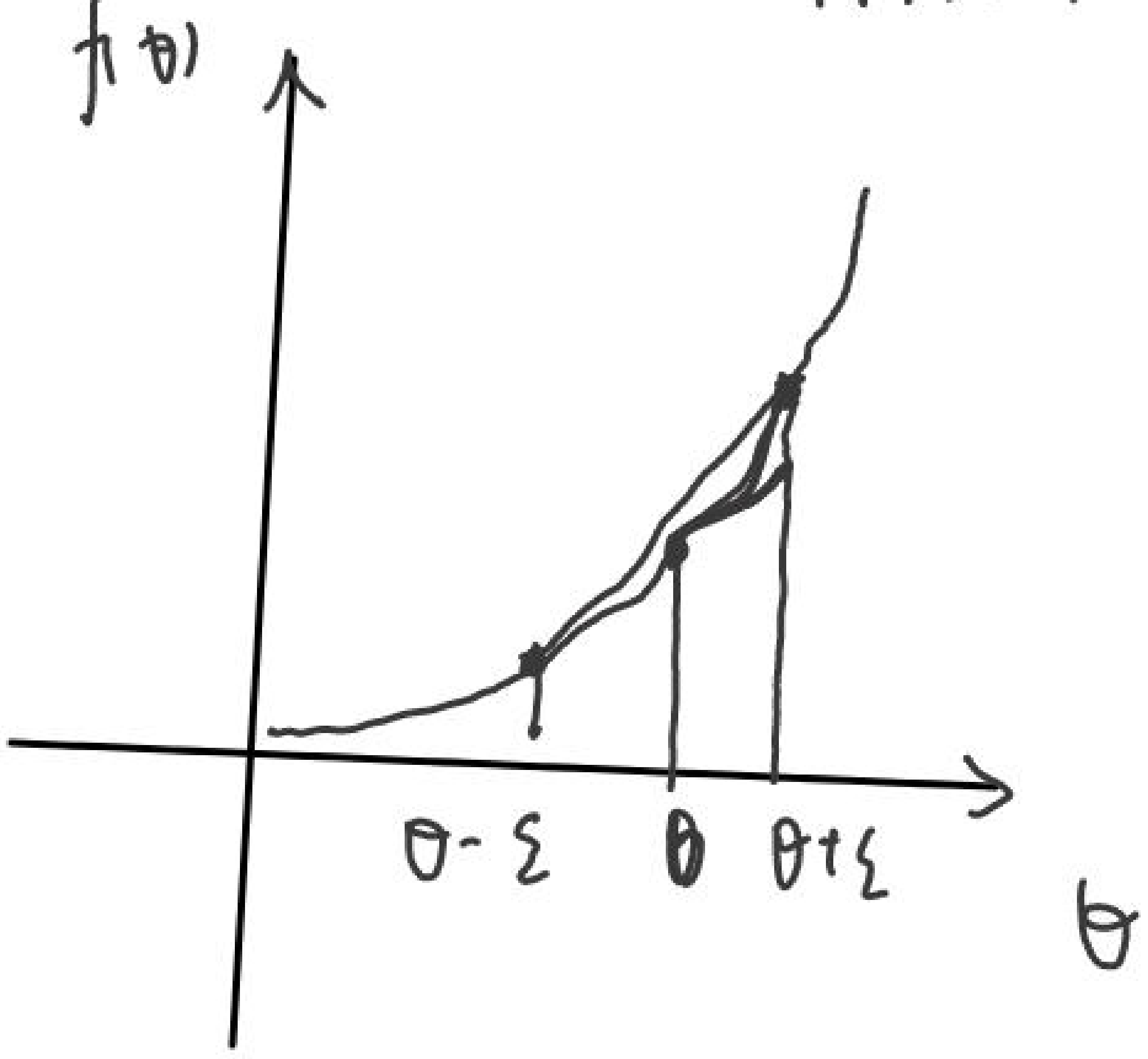
$$\therefore \quad \sigma^2 = \frac{1}{m}\sum_{i=1}^{m} x^{(i)2}$$

$$x = \frac{x}{\sigma^2}.$$



12 17

梯度估计：



$$f'(\theta) \approx \frac{f(\theta+\varepsilon) - f(\theta-\varepsilon)}{2\varepsilon} \sim O(\varepsilon^2)$$

$$\approx \frac{f(\theta+\varepsilon) - f(\theta)}{\varepsilon} \sim O(\varepsilon).$$

$$f(\theta+\varepsilon) = f(\theta) + f'(\theta)\varepsilon + \frac{1}{2!}f''(\theta)\varepsilon^2 + o(\varepsilon^2).$$

$$f(\theta+\varepsilon) = f(\theta) + f'(\theta)\varepsilon + o(\varepsilon).$$

$$f'(\theta) = \frac{f(\theta+\varepsilon) - f(\theta)}{\varepsilon} - o(\varepsilon)$$